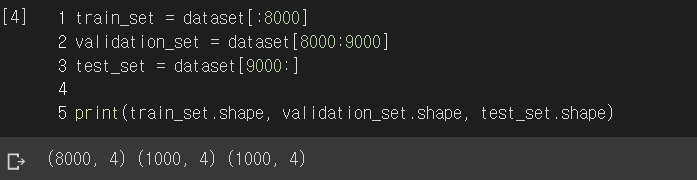
**CoE 202: Big Data Analysis and Machine Learning**

Homework 1: Regression for Syntactic data

20190146 Kim Yohan (김요한)

**#1. Splitting dataset**



**#2. Getting parameters to learn**

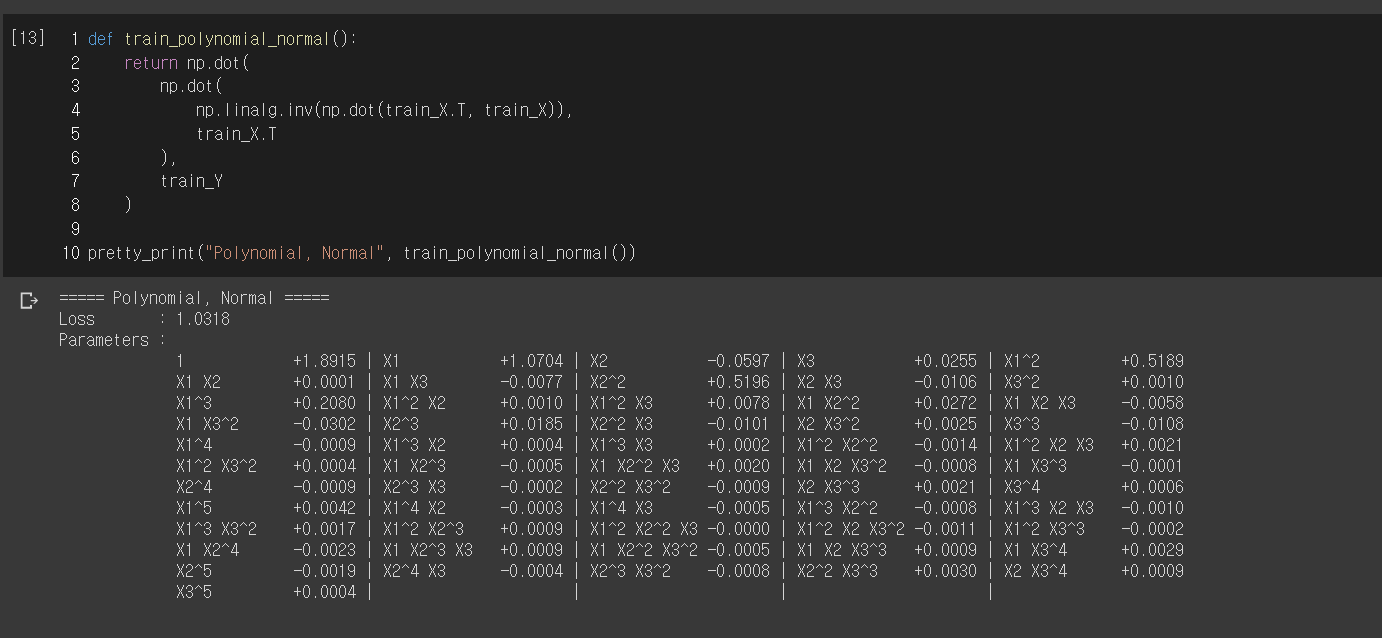
We need to train 56 parameters as we use degree 5 polynomial of 3 variables: x\_1, x\_2, x\_3.

It can also be obtained via PolynomialFeatures.get\_feature\_names.



**#3. Obtaining optimal parameter for plain Polynomial Regression**

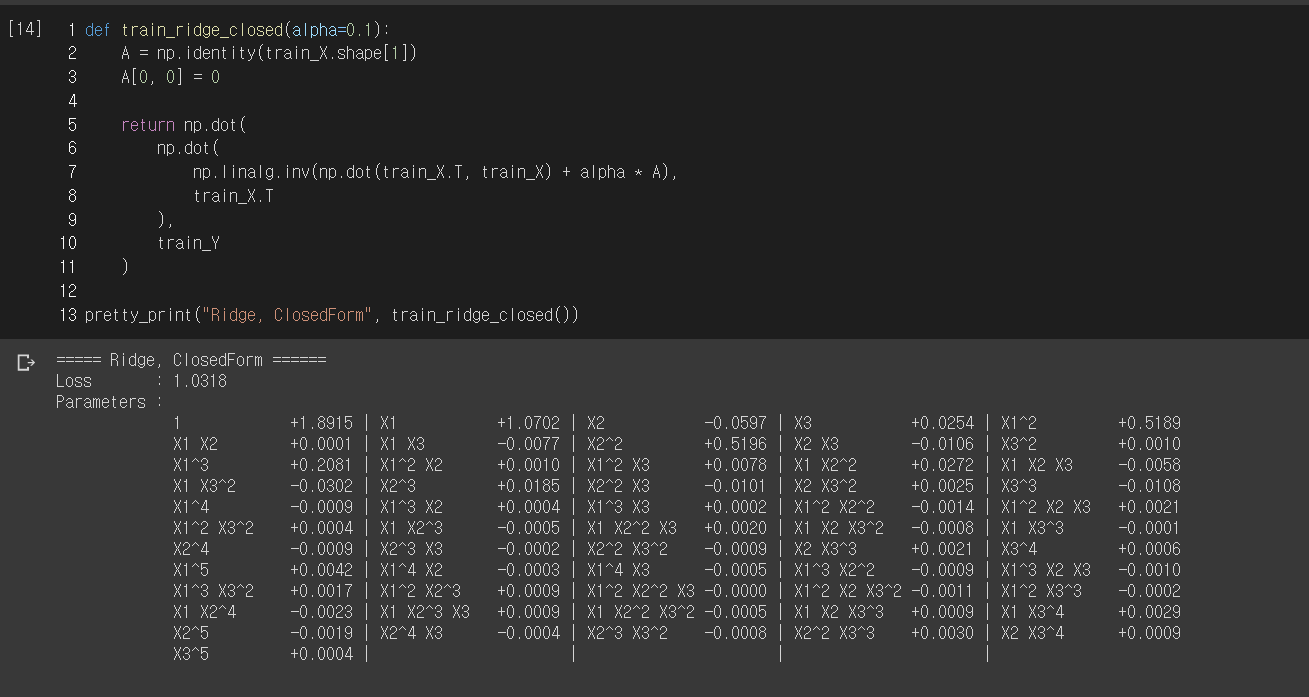
The optimal parameter can be obtained by directly evaluating the normal equation



The evaluated RMSE of test set is 1.0318, and the optimal parameters are shown in parameters table.

**#4. Obtaining optimal parameter for Ridge Regression**

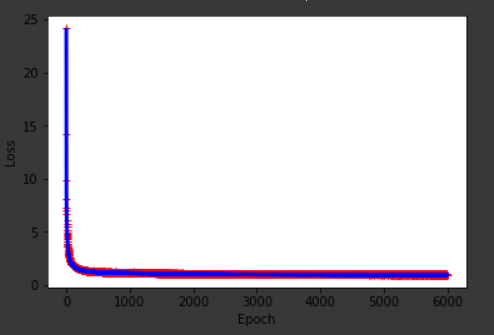
The optimal parameters for Ridge Regression can be obtained by evaluating the closed-form solution



The evaluated RMSE of test set is 1.0318, and the optimal parameters are shown in parameters table.

**#5. Training Plain Polynomial Regression by mini-batch Gradient Descent**

Hyperparameters are set as learning\_rate=0.000075 and epochs=6000

The learning curve, showing RMSE evaluated by train set and validation set against the epoch:

Blue: RMSE evaluated by the Validation Set

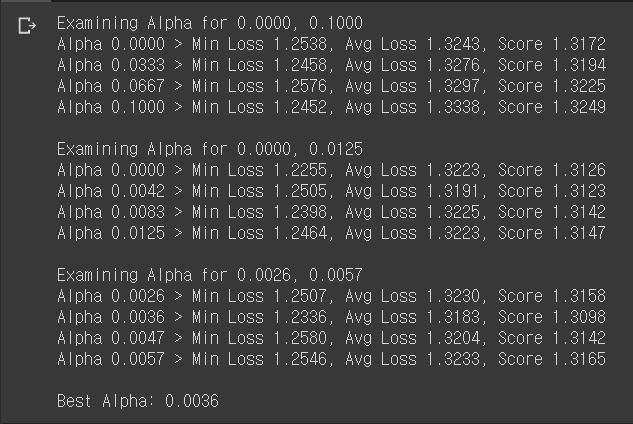
Red: RMSE evaluated by the Train Set

And the final RMSE value evaluated by the Test Set is 1.0496

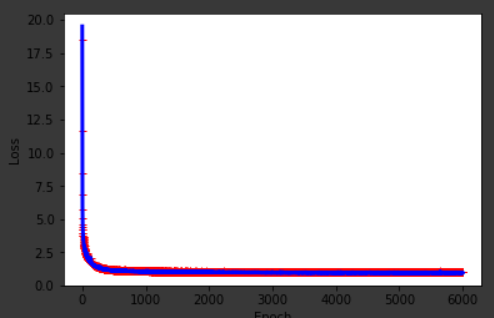
**#6. Training Ridge Regression by mini-batch Gradient Descent**

To adjust hyperparameter to get the best result, I defined a function running Scikit-Learn SGDRegressor w/ L2 regularization (learning\_rate=0.000005, epochs=6000) multiple time with different alpha values in [0, 0.1].

To justify, I have implemented cross-validation with 5 chunks and ran regressor total 30 times per each alpha value.



So, I have obtained best-performing alpha value 0.0036.

The learning curve is shown like this:

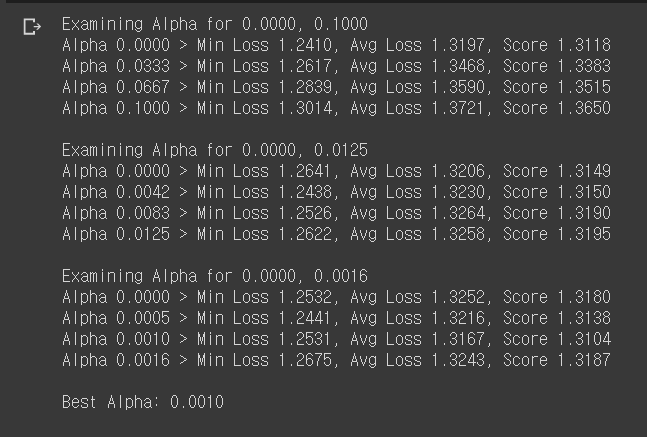
Blue: RMSE evaluated by the Validation Set

Red: RMSE evaluated by the Train Set

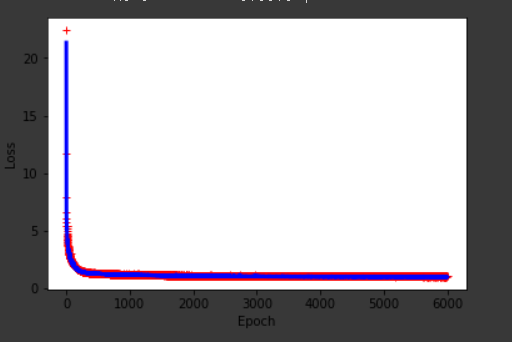
And the final RMSE value evaluated by the Test Set is slightly lower: 1.0476

**#7. Training Lasso Regression by mini-batch Gradient Descent**

Using same function defined in 6, but with L1 regularization, I have obtained the best-performing alpha value: 0.0010



The learning curve is shown like this:



Blue: RMSE evaluated by the Validation Set

Red: RMSE evaluated by the Train Set

And the final RMSE value evaluated by the Test Set is 1.0483

**#8. The Result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | Polynomial Regression  (Normal Equation) | Ridge Regression  (Closed-form Solution) | Polynomial Regression  (Minibatch GD) | Ridge Regression  (Minibatch GD) | Lasso Regression  (Minibatch GD) |
| Validation RMSE | 1.0318 | 1.0318 | 1.0496 | 1.0476 | 1.0483 |

As the Closed-form Solution by Polynomial Regression and Ridge Regression gave the least square error, the RMSE 1.0318, which is inevitable error from the noise N(0, 1).

When the Gradient Descent is used, the Ridge Regression showed the best result. The next is Lasso Regression and Plain Polynomial Regression showed the worst result among those.

I think it is because we used 5 degrees polynomial of 3 variables. By that, our model obtained more complexity and by the regularization our model tried to minimize total parameters while minimizing the error. I think it lead to minimize useless variable and make our model learn well.